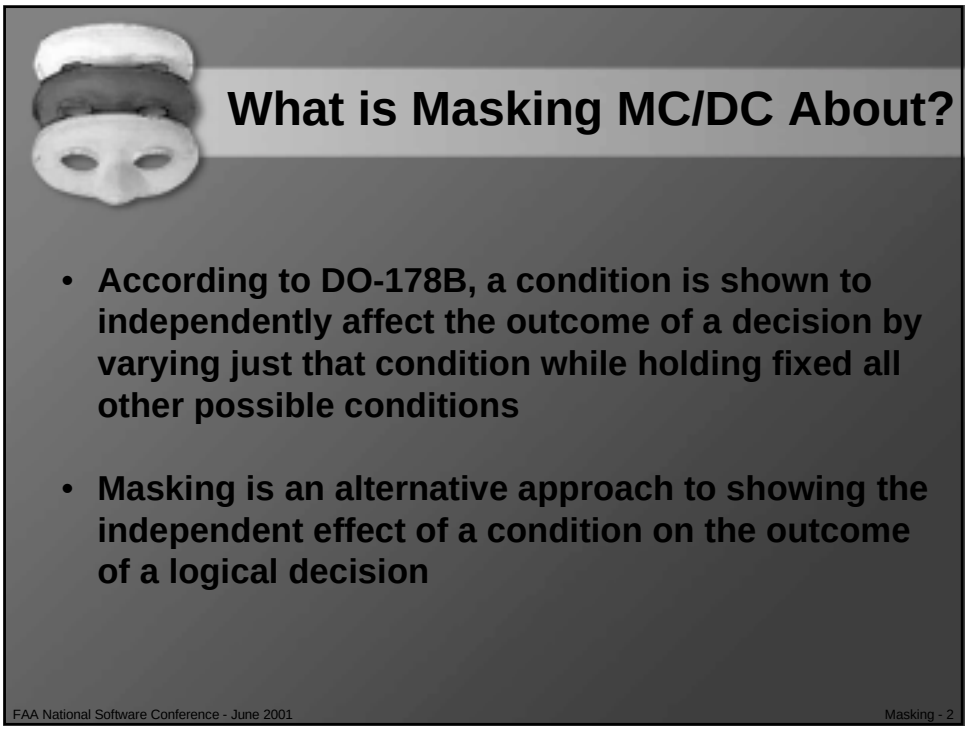
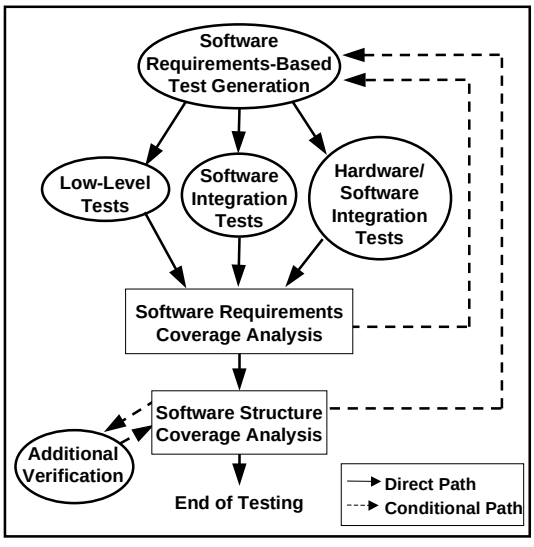
FAA National Software Conference, June 2001
MC/DC Masking



Outline

- **Structural Coverage in the Context of DO-178B**
- **Modified Condition/Decision Coverage (MC/DC)**
  - what it is
  - what it isn't
- **Different Approaches to MC/DC**
  - Unique Cause vs. Masking
    - how are they different & how are they the same
      - for expressions with common logical operators
        e.g., **A *or* B *or* C *or* D**; **A *and* B *and* C *and* D**
      - for expressions with mixed logical operators
        e.g., (**A *or* B**) ***and*** (**C *or* D**); (**A *and* B**) ***or*** (**C *xor* D**)

The Role of Structural Coverage

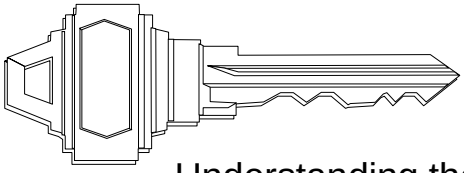


- Structural coverage analysis confirms that "the requirements-based test procedures exercised the code structure"
- Types of structural coverage in DO-178B
  - statement coverage (A-C)
  - decision coverage (A-B)
  - MC/DC (A)

Describing MC/DC

- **MC/DC is**
  - based on the following criteria:
    - ① every point of entry & exit in the program has been invoked at least once
    - ② every condition in a decision in the program has taken all possible outcomes at least once
    - ③ every decision in the program has taken all possible outcomes at least once
    - ④ each condition in a decision has been shown to independently affect that decision's outcome
- **MC/DC is *not***
  - a testing method
  - concerned with test cases developed from the source code (i.e., structural testing)
  - guaranteed at the source code level if measured at the object code (and vice versa)
    - MC/DC can be demonstrated at the object code level *if* analysis demonstrates that coverage at the object code will be equivalent to the same coverage at the source code (FAQ 42, DO-248A)

FAA National Software Conference - June 2001 Masking - 5



Understanding the implications of criteria ④ is the key to determining whether any approach (including masking) is acceptable for meeting the MC/DC objective

- What does independent effect mean?
- Why do it?

FAA National Software Conference - June 2001 Masking - 6

Independent Effect

- **A condition independently affects a decision's outcome if that condition *alone* determines the outcome of the decision**

a condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions

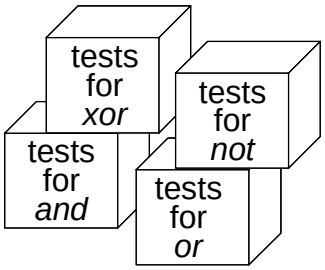
This tells you specifically HOW to show independent effect

- **Chilenski/Miller defined specific minimum tests to demonstrate the independent effect of each condition at individual logical operators**

Minimum Tests

- **The minimum tests are intended to assure that each input to a logical operator correctly affects the outcome**
- **The minimum tests provide the building blocks for assessing MC/DC**

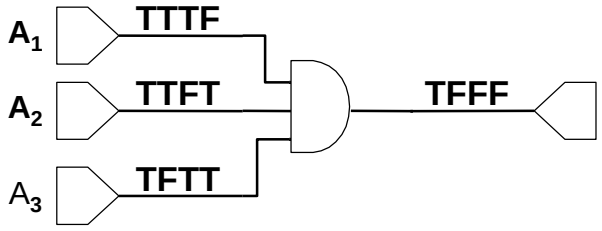
logical operator = logical gate



Testing an *n*-input *and* Gate

A_1 *and* A_2 *and* A_3 *and* … A_n

- **Minimum testing to provide MC/DC requires**
  - all inputs *true*, output *true*
  - each input individually *false*, output *false*
- **Example: testing a 3-input *and* gate requires TTT, TTF, TFT, FTT**

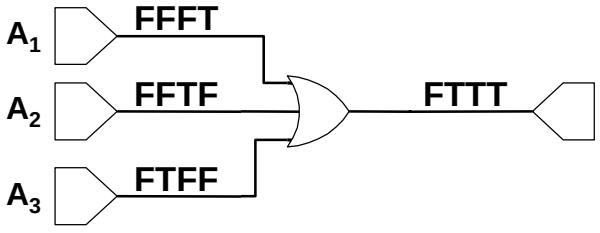


A_1 *and* A_2 *and* A_3

Testing an *n*-input *or* Gate

A_1 *or* A_2 *or* … A_n

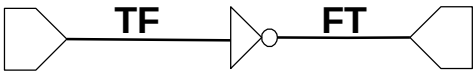
- **Minimum testing to provide MC/DC requires**
  - all inputs *false*, output *false*
  - each input individually *true*, output *true*
- **Example: testing a 3-input *or* gate requires FFF, FFT, FTF, TFF**



A_1 *or* A_2 *or* A_3

Testing a *not* Gate

- **Minimum testing to provide MC/DC requires**
  - input *true*, output *false*
  - input *false*, output *true*
- **Example:**



***not* A**

Testing an *xor* Gate

***xor* gates: are not like other gates**

- **More than one test set will satisfy the MC/DC criteria for an *xor* gate**
- **Minimum testing to provide MC/DC requires**
  - any of the following for a 2-input ***xor***
    - TT, TF, FT
    - TF, FT, FF
    - FT, FF, TT
    - FF, TT, TF

Controllability & Observability

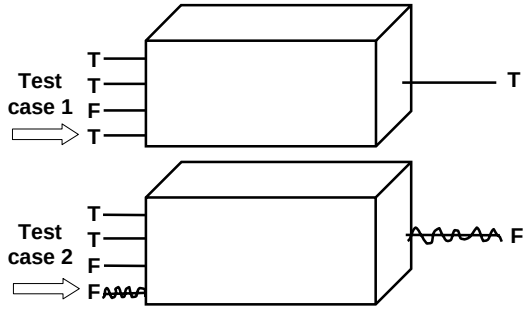
- **Basic concepts of testing logic circuits:**
  - *controllability:* ability to control the inputs to a logical operator
  - *observability:* ability to observe the outputs of a logical operator at some end point
- **The minimum tests establish the inputs and expected outputs needed at a logical operator to show independent effect of each condition**

Approaches to Independent Effect

- ***Unique cause* and *masking* are two approaches to showing the independent effect of a condition for multiple logical operators within a decision**
- **For expressions with common logical operators, unique cause and masking are the same**
  - for **A *or* B *or* C *or* D** or **A *and* B *and* C *and* D**
- **Differences emerge for expressions with mixed logical operators**
  - such as (**A *or* B**) ***and*** (**C *or* not D**)

Unique Cause

- **A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions**



(Assume that tests come from the requirements)

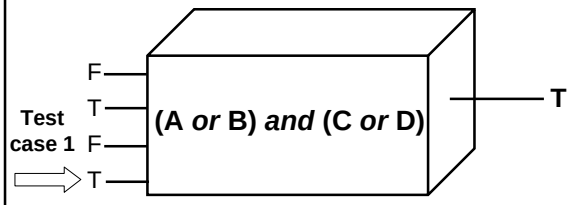
If only one input toggles and the outcome toggles in response, then the cause is assumed to be the toggled input

- don't need to see the internal logic of the expression to show independent effect

Masking

- **A condition is shown to independently affect a decision's outcome using logic principles to assure that no other condition influences the outcome**
  - even though more than one condition may change value
- **Some inputs may hide or mask the effect of other inputs**
  - *false* ***and*** **X** is always *false*
  - *true* ***or*** **X** is always *true*
- **"Masking" principles are the converse**
  - *true* ***and*** **X** is **X**
  - *false* ***or*** **X** is **X**

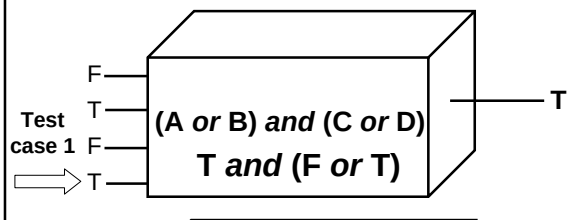
Masking



Analysis of the internal logic is needed to show that the condition of interest is the *only* toggled condition causing the decision's outcome to toggle.

FAA National Software Conference - June 2001 Masking - 17

Masking (cont.)



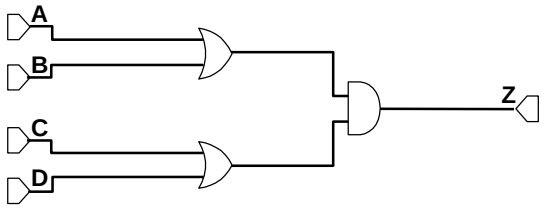
Analysis of the internal logic is needed to show that the condition of interest is the *only* toggled condition causing the decision's outcome to toggle.

D is the only input that affects the outcome

FAA National Software Conference - June 2001 Masking - 18

Example Z:= (A or B) and (C or D);

Requirements:



Requirements-based test cases:

	1	2	3	4	5
A	F	F	T	T	F
B	F	T	F	F	T
C	T	T	T	F	F
D	F	F	F	F	T
Z	F	T	T	F	T

Do these test cases provide MC/DC of the source code?

Source code: Z:= (A or B) and (C or D);

Truth Table Approach

Z:= (A or B) and (C or D);

A	B	C	D	Z
F	F	T	F	F
F	T	T	F	T
T	F	T	F	T
T	F	F	F	F
F	T	F	T	T

- Look for pairs of test cases where only one input value changes -- and the outcome changes
- There are no pairs of test cases where **D** is the only input value that changes

Truth Table Approach

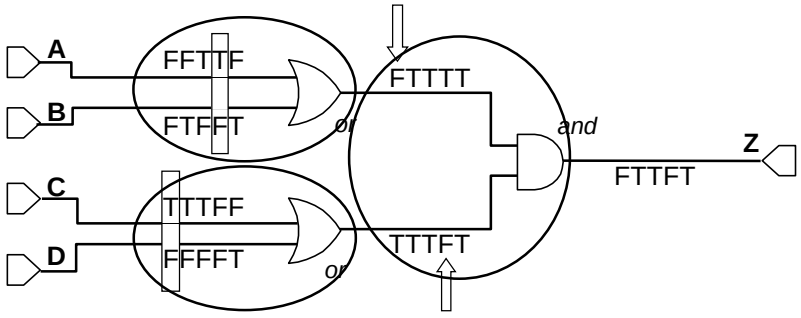
Z:= (A or B) and (C or D);

A	B	(A or B)	C	D	(C or D)	Z
F	F	F	T	F	T	F
F	T	T	T	F	T	T
T	F	T	T	F	T	T
T	F	T	F	F	F	F
F	T	T	F	T	T	T

- In these 2 test cases, **D** is the *only* condition that causes the outcome to change
  - these 2 cases show the independent effect of **D** -- even though more than one condition changes value

FAA National Software Conference - June 2001 Masking - 21

Check for Minimum Tests



- **Check for observability -- are the outputs of the *or* gates observable?**
- **Check for controllability -- do minimum tests exist for each logical gate?**

FAA National Software Conference - June 2001 Masking - 22

Analysis to confirm that you have the minimum tests is required for masking -- as opposed to simply showing independence pairs

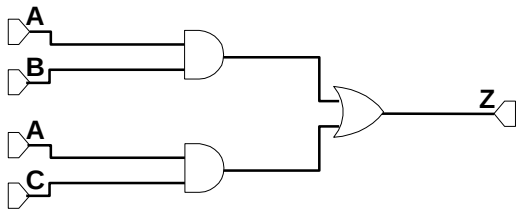
Coupled Conditions

One condition is coupled with another condition if the value of one condition influences the value of the other

- **A test set for an expression with strongly coupled conditions *cannot* meet MC/DC using the unique cause approach**
- **A test set for an expression with coupled conditions *may* meet MC/DC using the masking approach**

Example Z := (A and B) or (A and C);

Requirements:



Requirements-based test cases:

	1	2	3	4	5
A	T	F	T	T	F
B	T	T	F	F	F
C	F	F	F	T	T
Z	T	F	F	T	F

Do these test cases provide MC/DC of the source code?

Source Code: Z:= (A and B) or (A and C);

Truth Table Approach

Z:= (A and B) or (A and C);

A	B		A	C		Z
T	T		T	F		T
F	T		F	F		F
T	F		T	F		F
T	F		T	T		T
F	F		F	T		F

- Expand the test cases to account for **A** being treated as 2 distinct conditions

Truth Table Approach

Z:= (A and B) or (A and C);

A	B	A and B	A	C	A and C	Z
T	T	T	T	F	F	T
F	T	F	F	F	F	F
T	F	F	T	F	F	F
T	F	F	T	T	T	T
F	F	F	F	T	F	F

- Expand the test cases to account for **A** being treated as 2 distinct conditions

⇨ Add the value of the subterms (**A *and* B**) and (**A *and* C**)

Truth Table Approach

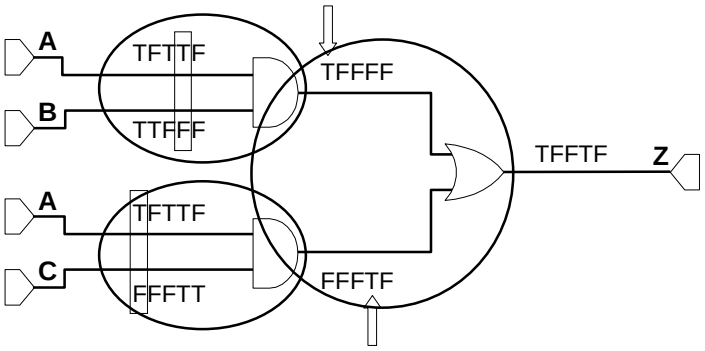
Z:= (A and B) or (A and C);

A	B	A and B	A	C	A and C	Z
T	T	T	T	F	F	T
F	T	F	F	F	F	F
T	F	F	T	F	F	F
T	F	F	T	T	T	T
F	F	F	F	T	F	F

- Expand the test cases to account for **A** being treated as 2 distinct conditions
- Add the value of the subterms (**A *and* B**) and (**A *and* C**)

⇨ Check for independence pairs

Check for Minimum Tests



- **Check for observability -- are the outputs of the *and* gates observable?**
- **Check for controllability -- do minimum tests exist for each logical gate?**

Pros and Cons of Masking

Pros	Cons
• **Applies to more logic expressions than unique cause** – because masking applies to expressions with coupled conditions	• **Requires analysis of the logic of each decision (this is not required for unique cause)**
• **Provides an additional check on the correctness of the source code**	• **Requires visibility into the logic of the source code**
• **Provides a practical approach for confirming MC/DC – both for manual and automated projects**	

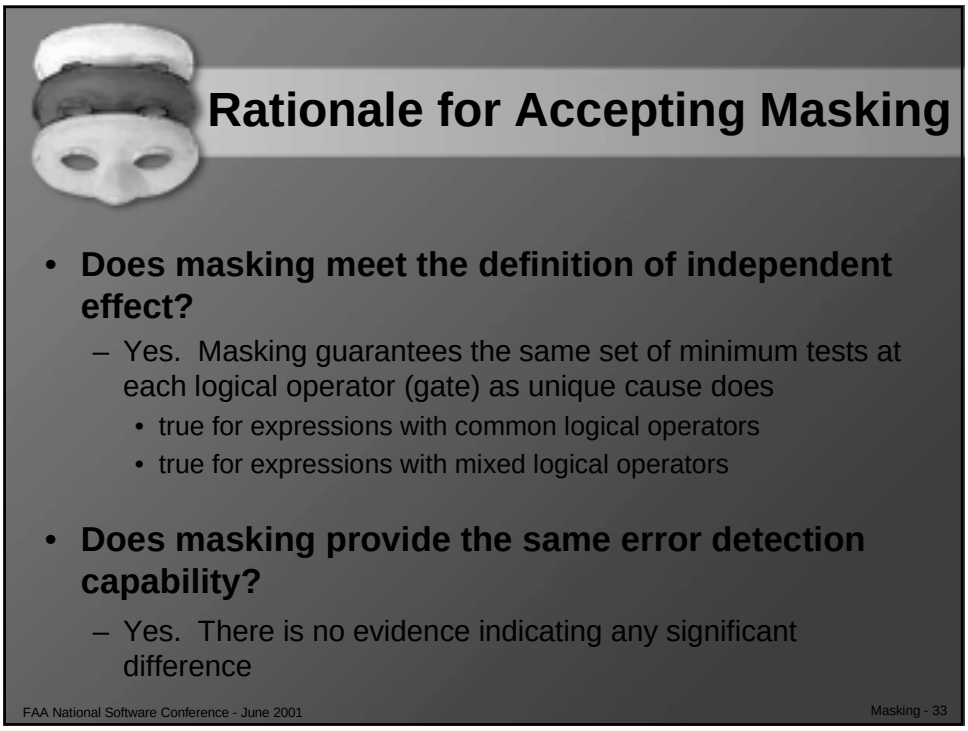
Bottom Line

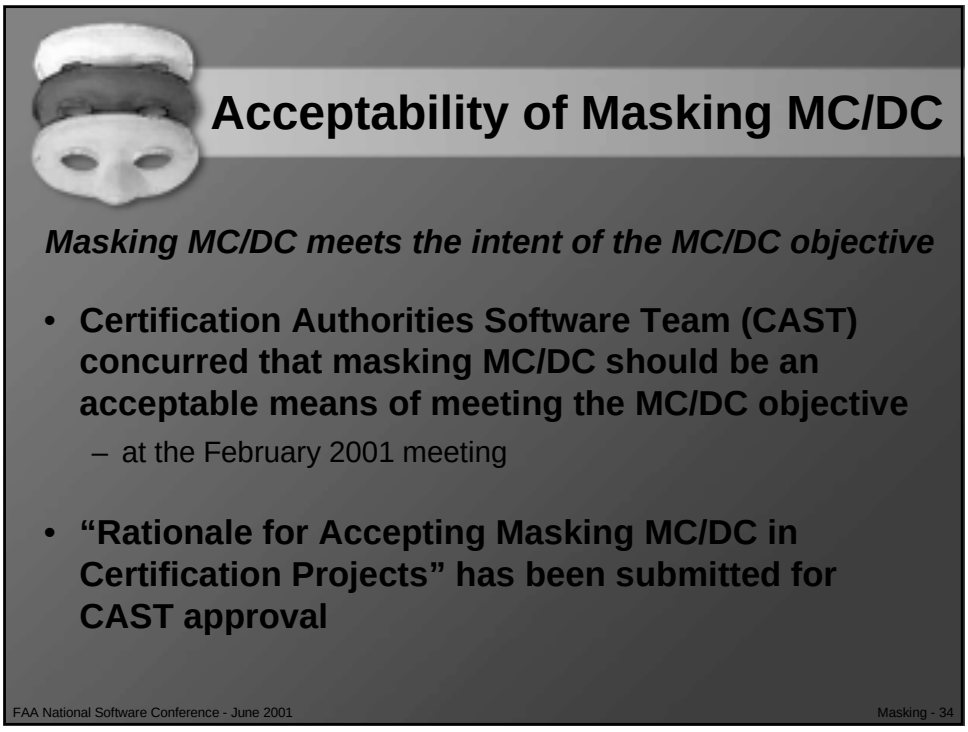
- **You lose *nothing* using masking**
  - *however*, the masking approach requires analysis of the logic of each expression (that is not required for unique cause) to confirm the independent effect of each condition
- **You *gain* a method to handle expressions with coupled conditions**

Error Detection

Initial concern: masking would detect fewer errors than unique cause

- **For expressions with common logical operators**
  - there is no difference in error detection between unique cause and masking
    - *because* there is no difference in the minimum test sets
- **For expressions with mixed logical operators**
  - no evidence exists of a practical difference in error detection between unique cause and masking
  - Chilenski's analysis of error sensitivity between unique cause and masking "has not shown that there is any significant difference."


Rationale for Accepting Masking
• Does masking meet the definition of independent effect?
– Yes. Masking guarantees the same set of minimum tests at each logical operator (gate) as unique cause does
• true for expressions with common logical operators
• true for expressions with mixed logical operators
• Does masking provide the same error detection capability?
– Yes. There is no evidence indicating any significant difference
FAA National Software Conference - June 2001
Masking - 33


Acceptability of Masking MC/DC
Masking MC/DC meets the intent of the MC/DC objective
• Certification Authorities Software Team (CAST) concurred that masking MC/DC should be an acceptable means of meeting the MC/DC objective
– at the February 2001 meeting
• "Rationale for Accepting Masking MC/DC in Certification Projects" has been submitted for CAST approval
FAA National Software Conference - June 2001
Masking - 34